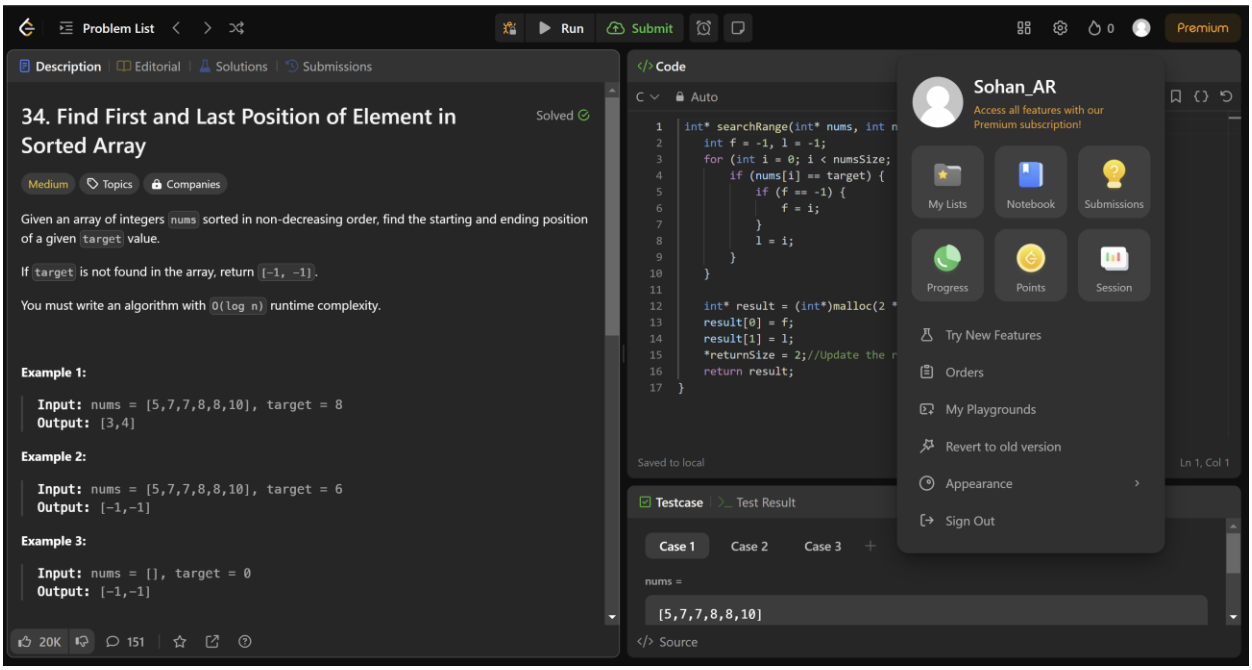
```
/**
 * Note: The returned array must be malloced, assume caller calls free().
 */
 int* searchRange(int* nums, int numsSize, int target, int* returnSize) {
   int f = -1, l = -1;
   for (int i = 0; i < numsSize; i++) {
      if (nums[i] == target) {
         if (f == -1) {
            f = i;
         }
         l = i;
      }
   }

   int* result = (int*)malloc(2 * sizeof(int));
   result[0] = f;
   result[1] = l;
   *returnSize = 2;//Update the return size
   return result;
```

}


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
34. Find First and Last Position of Element in Sorted Array
Solved
Medium
Topics
Companies
Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value.
If target is not found in the array, return [-1, -1].
You must write an algorithm with O(log n) runtime complexity.
Example 1:
Input: nums = [5,7,7,8,8,10], target = 8
Output: [3,4]
Example 2:
Input: nums = [5,7,7,8,8,10], target = 6
Output: [-1,-1]
Example 3:
Input: nums = [], target = 0
Output: [-1,-1]
20K
151
Code
C
Auto
int* searchRange(int* nums, int n
int f = -1, l = -1;
for (int i = 0; i < numsSize;
if (nums[i] == target) {
if (f == -1) {
f = i;
}
l = i;
}
}
int* result = (int*)malloc(2 *
result[0] = f;
result[1] = l;
*returnSize = 2;//Update the r
return result;
}
Saved to local
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
nums =
[5,7,7,8,8,10]
Source
Sohan_AR
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Session
Try New Features
Orders
My Playgrounds
Revert to old version
Appearance
Sign Out